Deconvolution of Spectral Voigt Profiles Using Inverse Methods and Fourier Transforms

Genia Vogman
Senior Thesis

June 2010

Contents

1 Introduction

Emissive spectroscopy, which is a means of measuring light radiation emitted by high-energy particles, is the basis for many experimental studies. Through spectroscopy it is possible to determine particle temperature and density by decomposing emitted light into constituent wavelengths and resolving the shapes of individual spectral profiles. This is particularly useful for the study of stellar and terrestrial plasmas, in which particles can reach temperatures that are so high that their properties cannot be measured by any other means.

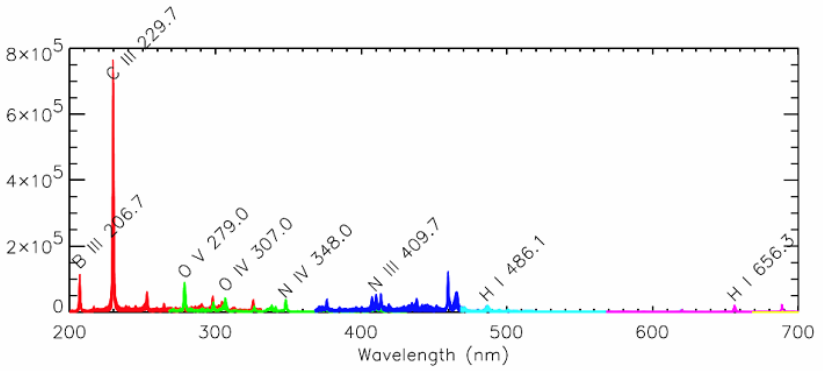


Figure 1: An emission light spectrum [9] of intensity, in arbitrary units, versus wavelength from 200 nm (ultraviolet) to 700 nm (infrared).

What makes spectroscopy particularly useful is the ready availability of data – that is, the retrieval of spectroscopy data is much easier than the use of other diagnostics. However, the caveat to this is that in order to obtain accurate measurements of temperatures and densities, light radiation data must be processed. Such processing involves correction of instrument-associated distortions followed by the separation of one physical effect from another – such as the separation of temperature effects from density effects. Instrument distortion, which arises from the inherent nature of light measurement and curved optics, often makes it difficult to accurately resolve physical effects. However, in the case of a fully-calibrated spectroscopic system, whose instrument effects are well known and can thus be separated from measured physical effects, it is possible to determine particle temperature and density.

To resolve temperature and density from light radiation data it is necessary to deconvolve individual spectral lines. An individual spectral line is well-approximated by a Voigt function, which is the convolution of an unnormalized Gaussian with an unnormalized Lorentzian. The full width at half maximum (FWHM) of a Gaussian is directly related to particle temperature, while the FWHM of a Lorentzian is directly proportional to particle density. Thus by deconvolving the two profiles and determining their FWHM, it is possible to resolve the two critical properties. Two techniques for performing such a deconvolution and resolving temperature and density are described; a direct inverse method and an inverse method incorporating Fourier transforms.

2 Light Spectrum and Individual Line Profiles

Light radiation emitted by particles is characterized by a set of discrete intensity peaks, or "lines," that are a function of wavelength. The wavelength at which a peak occurs is dictated by: the element, the ionization state, and the energy level transition of the electron that gives rise to a photon. Hence a typical emission spectrum looks like that shown in Figure 1, where individual peaks represent light emitted by a particular ion. In fact, every ion has a distinct spectral signature, which is how it is possible to identify them to begin with. Intensity is dictated by the number of photons captured by the optics and by instrument signal amplification. [9]

On a smaller scale, these spectra have a fine structure (see Figure 2) that is directly related to the physics of the electron energy level transition. In particular this structure is dictated by a number of effects whose general relative shapes are shown in Figure 3. A set of ions that are at a temperature of zero Kelvin and experience the same electron energy level transition will have an infinitesimally thin (resembling a delta function) spectral line profile at a nominal wavelength

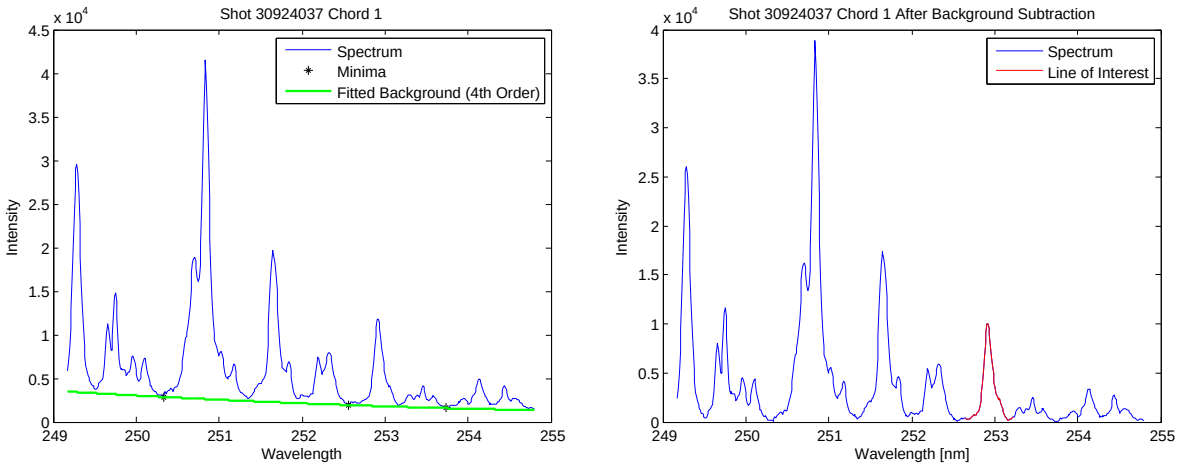


Figure 2: (a) A typical emission spectrum from a high-temperature plasma experiment. Shot 30924037 Chord 1 refers to the experimental pulse and the fiber optic chord (the first out of twenty) used to collect light radiation data. (b) Spectrum after background subtraction. In this case the 252.93 nm O III line is highlighted to distinguish it from other lines.

associated with the transition. Because instruments cannot capture infinitesimally thin signals, they cause the line profile to appear broader. This finite thickness associated with instrumentation must be accounted for through calibration before any physical effects can be resolved.

2.1 Gaussian Profile and Doppler Broadening

If ions are at a non-zero temperature, their profiles are widened further in accordance with the Doppler effect.[6] The Doppler effect is associated with temperature; particles moving in random directions due to thermal motion – away and towards the observing optics – will produce a Gaussian profile. The centroid of the Gaussian, that is the wavelength λ_0 at which the function is centered, is the nominal wavelength and its full width at half max (FWHM) is related to temperature. The Gaussian can be expressed as a function of wavelength λ as follows:

$$G(\lambda) = G_{max} \exp\left(-\frac{4\ln 2(\lambda - \lambda_0)^2}{G_w^2}\right) \tag{1}$$

where

$$G_w = \frac{2\lambda_{nom}}{c}\sqrt{\frac{2T\ln 2}{m_i}}, \tag{2}$$

G_{max} is the amplitude, λ_0 is the centroid wavelength seen by the instrument, G_w is the FHWM, λ_{nom} is the nominal centroid wavelength (derived empirically[10]) associated with the energy of the electron transition, T is temperature in eV (see Appendix A), c is the speed of light, and m_i is the mass of the radiating ion. In the case of the N II line shown in Figure 3, the empirically-derived[10] nominal wavelength (the centroid) is 306.283 nm.

2.2 Lorentzian Profile and Stark Broadening

In addition to temperature broadening, the Stark effect also causes lines to broaden. The Stark effect results from the electric field imposed on the radiating particle by the charged particles

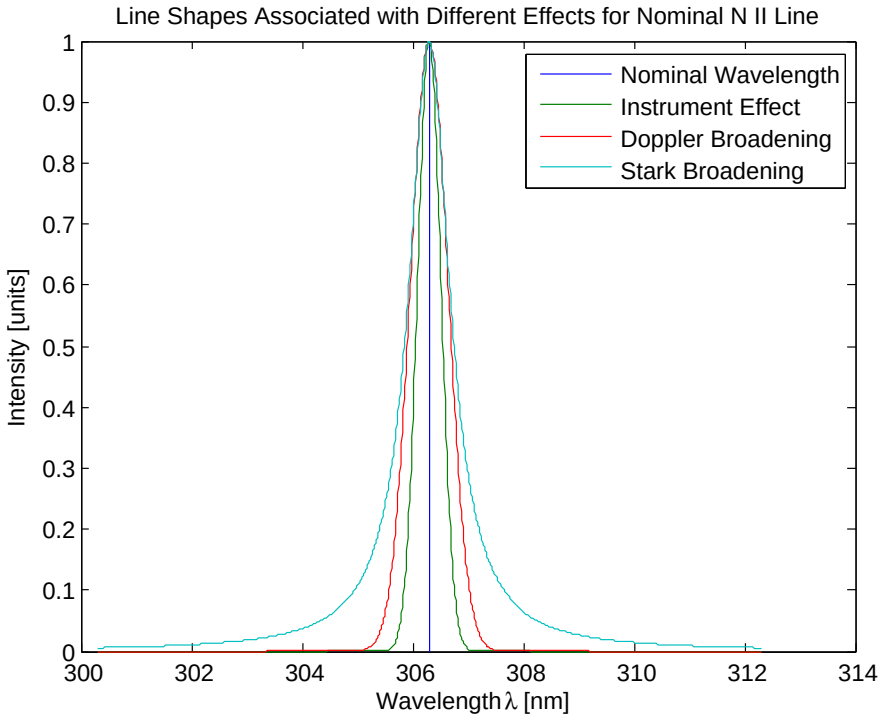


Figure 3: Mechanisms that can influence line shape are: instrument effects associated with optical limitations, Doppler broadening associated with particle temperature, and Stark broadening associated with particle density.

surrounding it and causes spectral lines to have a Lorentzian profile, as that seen in Figure 3. The Lorentzian profile is described by the following function[5]:

$$L(\lambda) = L_{max} \frac{L_w^2}{4\left(\lambda - \lambda_0\right)^2 + L_w^2}, \tag{3}$$

where L_{max} is the amplitude, L_w is the FWHM, and λ_0 is the centroid. The way in which the FWHM of the Lorentzian is defined will depend on the type of atoms involved and the approximations used in arriving at the quantum mechanical description, which will not be described here. For the case of non-hydrogenic atoms, L_w is related to electron density[5] as follows

$$L_w = (2 \times 10^{-18}) w n_e + (3.75 \times 10^{-4}) n_e^{1/4} \alpha (1 - 0.068 n_e^{1/6} T_e^{-1/2}) \tag{4}$$

where w is the electron impact parameter, n_e is the electron density in cm^{-3}, α is the static ion broadening parameter, T_e is the electron temperature in Kelvin, and the resulting L_w is in nanometers. The parameters w and α are tabulated quantities.[7] The first term on the right hand side of Equation 4 is associated with broadening due to electron electric field contribution and the second term is the ion correction factor. For ions that are non-hydrogenic Stark broadening is predominantly governed by electron effects. As such the second term in Equation 4 can be neglected[11] and consequently

$$L_w = (2 \times 10^{-18}) w n_e. \tag{5}$$

Notably both the Gaussian and Lorentzian line profiles are not normalized since the signal amplitude is case-dependent and based on the number of emitters and instrument amplification. This means that in the case of the spectral profiles discussed, statistical distributions hold no meaning.

2.3 Other Profile-Changing Effects

There are other effects that can influence the shape of a spectral line, but these are often negligible (such as natural broadening[6]) or are easily resolved. As an example, Bremsstrahlung radiation, radiation caused by free electrons accelerating in the presence of other charged particles, is one type of radiation effect that can distort the base of ion emission lines. Because Bremsstrahlung radiation is associated with electrons and not ions, however, it does not affect individual ion line profiles and consequently can be easily subtracted as the background radiation as seen in Figure 2.

2.4 Convolution of Gaussian and Lorentzian Functions

When there are multiple broadening effects the resulting line profile is the convolution of the constituent profiles.[6] For the case of the Lorentzian and Gaussian, the convolution is described by a Voigt function[8]:

$$V(\lambda) = [G \otimes L](\lambda) = \int_{-\infty}^{\infty} G(\lambda')L(\lambda - \lambda')d\lambda' \tag{6}$$

$$= \int_{-\infty}^{\infty} G_{max} \exp\left(-\frac{4\ln 2(\lambda' - \lambda_0)^2}{G_w^2}\right) L_{max} \frac{L_w^2}{4\left([\lambda - \lambda'] - \lambda_0\right)^2 + L_w^2} d\lambda' \tag{7}$$

$$= \int_{-\infty}^{\infty} G_{max} \exp\left(-\frac{4\ln 2(\lambda' - \lambda_0)^2}{G_w^2}\right) L_{max} \frac{L_w^2}{4\left([\lambda - \lambda'] - \lambda_0\right)^2 + L_w^2} d\lambda' \tag{8}$$

$$= A \int_{-\infty}^{\infty} \exp\left(-\frac{4\ln 2(\lambda' - \lambda_0)^2}{G_w^2}\right) \frac{L_w^2}{4\left([\lambda - \lambda'] - \lambda_0\right)^2 + L_w^2} d\lambda'. \tag{9}$$

where A denotes the effective amplitude of the Voigt profile.

For a spectroscopic system in which instrument effects are well known, Doppler and Stark broadening are the paramount contributors to spectral line shape.[5;6] As such, isolated spectral lines are well-described by the Voigt function. This means that if one spectral line is to be fitted with a Voigt profile, it should be sufficiently isolated so that neighboring lines do not contaminate the profile. Contaminated profiles are seen in Figure 2, where many neighboring lines overlap and create complicated structures.

3 Direct Inverse Method for Resolving Temperature and Density

The primary objective of exploring the structure of spectral lines, at least for the purposes of this paper, is to resolve physical effects like temperature and density. In order to resolve these two parameters from a given isolated spectral line, it is necessary to fit the line with a Voigt function and from the best fit deduce the optimal T and n_e.

3.1 Recovering Temperature and Density from a Voigt Fit

A single spectral line consists of data in the form of coordinate points – wavelength $\{\lambda_i\}$ and intensity $\{I_i\}$. Because the amplitude of the peak can be quite high, and because it is independent of the other parameters, the raw spectral data is initially normalized to an intensity peak of one (as opposed to an integral of one) so that $I_{max} = 1$. This is done for convenience purposes only and has no effect on the FWHM of a given profile or its constituents. In order to fit the set of points defined by $\{\lambda_i\}$ and $\{I_i\}$, an algorithm has been constructed. This algorithm incorporates the following steps:

1. Construct a vector λ_i' which is bounded by the minimum and maximum of $\{\lambda_i\}$, but has a finer resolution (0.005 nm instead of ≈0.1 nm).

2. Construct a function of unknowns that can be fed into an optimization algorithm

   (a) Express Gaussian and Lorentzian in terms of unknowns u_1, u_2, and u_3 which correspond to temperature T, density n_e, and the centroid λ_0 respectively:

$$G(\lambda') = \exp\left(-\frac{4\ln 2(\lambda' - u_3)^2}{\left(\frac{2\lambda_{nom}}{c}\sqrt{\frac{2u_1 \ln 2}{m_i}}\right)^2}\right) \tag{10}$$

$$L(\lambda') = \frac{\left(2 \times 10^{-18} w u_2\right)^2}{4\left(\lambda' - u_3\right)^2 + \left(2 \times 10^{-18} w u_2\right)^2} \tag{11}$$

   (b) Convolve the profiles according to the discretized form of the convolution seen in Equation 6 and let u_4 be the amplitude A of this convolution and let $\Delta\lambda' = \lambda'_{i+1} - \lambda'_i$:

$$V(k) = u_4 \sum_j G(j) L(k - j + 1)\Delta\lambda', \tag{12}$$

   (c) Interpolate V onto the domain defined by λ and call this V_{fit} – the fitted Voigt profile.

   (d) Compute the least squares error E

$$E = \sum_{i=1}^{n} (V_{fit,i} - I_i)^2 \tag{13}$$

3. Minimize error E in the function described above by applying the Nedler-Mead simplex method[14] to optimize unknowns u_1, u_2, u_3, and u_4.

4. The output is then the temperature T (u_1), the density n_e (u_2), the centroid λ_0 (u_3), and the amplitude (A u_4). The first two parameters are the ones we sought.

Note that the amplitude term, one of the optimized unknowns (u_4), is the product of G_{max} and L_{max} as well as other constants. However, there is no need to resolve G_{max} and L_{max} as they are completely independent of the critical parameters of temperature T and density n_e.

A fit generated by the algorithm described above is shown in Figure 4. Notably this fit has no weighting associated with it. With weights, the wings of the profile can be fitted slightly better but the overall fit is considerably worse (see Figure 5). Since weighting is somewhat arbitrary, it will not be utilized or alluded to in further discussion; Figure 5 is meant purely to illustrate the flexibility of the algorithm.

3.2 Parameter Search to Verify Best Fit

In order to determine the nature of the found minimum, a search of the parameter space is performed. Another algorithm was constructed (see Appendix Section B.1) to determine the least squares error between the spectral data and the Voigt profile generated using a range of temperatures and densities. This algorithm was constructed as follows:

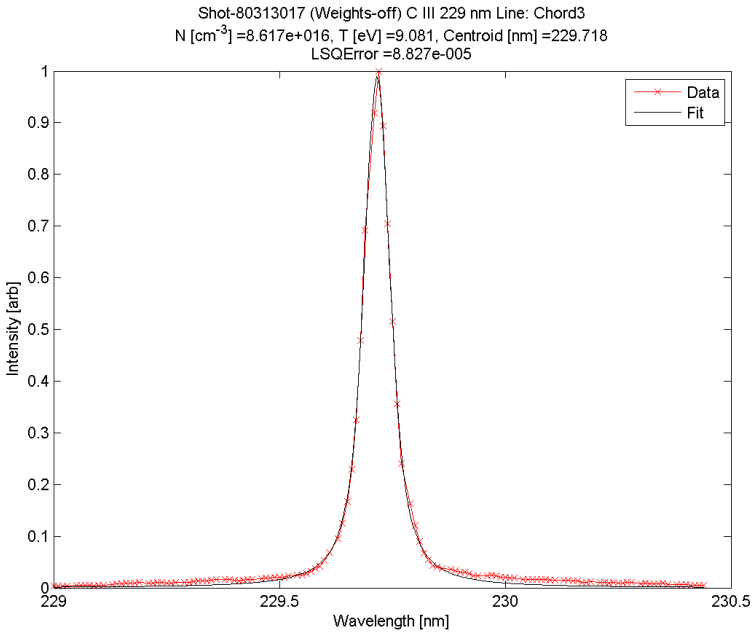


Figure 4: A Voigt fit using the direct inverse method algorithm produced the sought-after quantities of temperature and density, which are listed above the plot. The spectral data consists of 128 points and is plotted in red, while the Voigt function that was fit to it is in black.

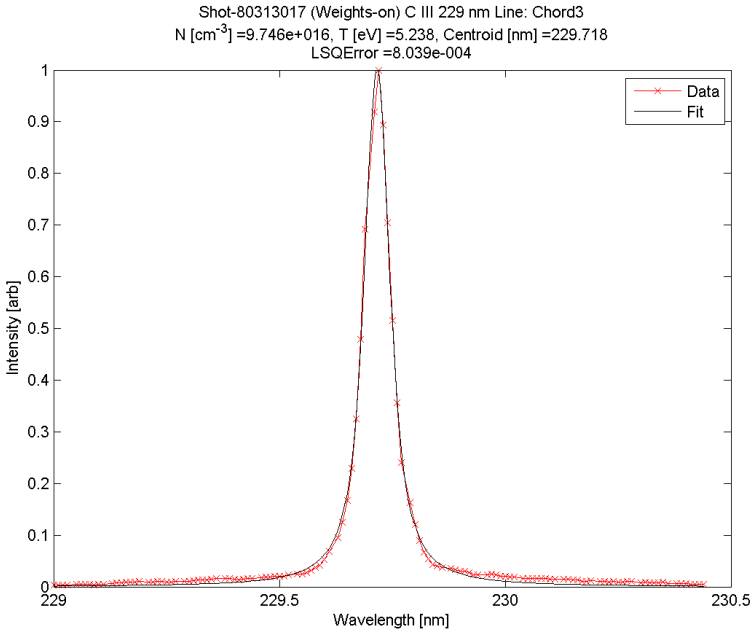


Figure 5: A weighted Voigt fit using the direct inverse method produced the sought-after quantities of temperature and density, which are listed above the plot. The peripherals of the Voigt profile are weighted higher than the center peak.

1. Select a range of temperatures and densities that are reasonable for the experiment of interest. In this case the temperature range is between 1 and 100 eV, and the density range is between 1×10^{22} and 15×10^{22} particles per cm^3.

2. For each combination of temperatures and densities do the following:

   (a) Express Gaussian and Lorentzian in terms of unknowns u_1, u_2, and u_3 which correspond to temperature T, density n_e, and the centroid respectively:

$$G(\lambda') = \exp\left(-\frac{4\ln 2(\lambda' - u_3)^2}{\left(\frac{2\lambda_{nom}}{c}\sqrt{\frac{2u_1\ln 2}{m_i}}\right)^2}\right) \tag{14}$$

$$L(\lambda') = \frac{\left(2\times10^{-18}wu_2\right)^2}{4\left(\lambda' - u_3\right)^2 + \left(2\times10^{-18}wu_2\right)^2} \tag{15}$$

   (b) Convolve the profiles according to the discretized form of the convolution seen in Equation 6 and let u_4 be the amplitude A of this convolution and let $\Delta\lambda' = \lambda'_{i+1} - \lambda'_i$:

$$V(k) = u_4\sum_j G(j)L(k-j+1)\Delta\lambda', \tag{16}$$

   (c) Interpolate V onto the domain defined by λ and call this V_{fit} – the fitted Voigt profile.

   (d) Compute the least squares error E

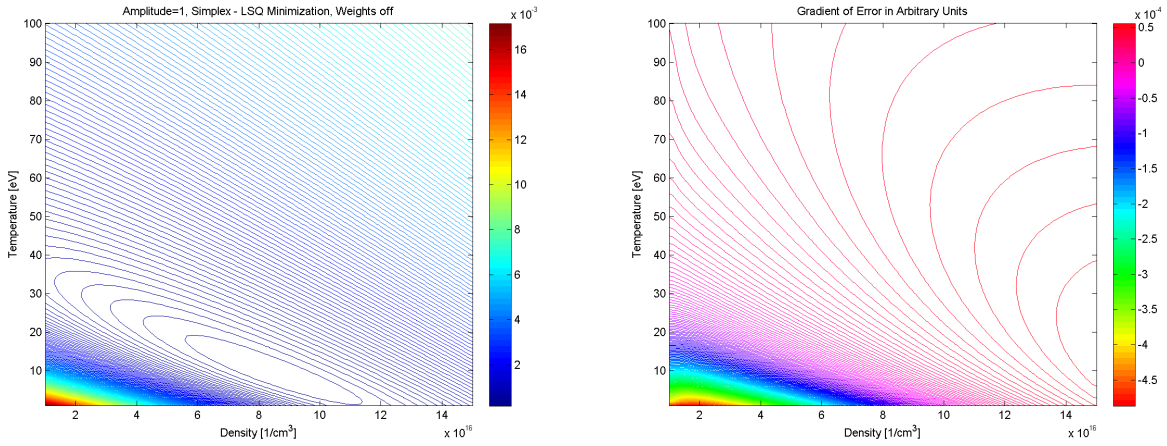
$$E = \sum_{i=1}^{n}(V_{fit,i} - I_i)^2 \tag{17}$$

3. The output of the algorithm is then a matrix of least squares errors for every combination of temperature T and density n_e.

By computing the error for a range of temperatures and densities it is possible to make certain deductions about the minimum found using the direct inverse method algorithm. For instance, it is possible to confirm whether the optimization routine was successful in finding a minimum, whether it is a shallow minimum, and whether it is the sole minimum. A contour plot of error versus temperature and density values is shown in Figure 6a. The plot clearly indicates that there is only one global minimum and it is in a region with a very small gradient (see Figure 6b).

In addition to identifying the region with the global minimum, it is useful to compare the parameter search minimum to the minimum found by the fitting algorithm. Figure 7 shows where the optimal solution for T and n_e, that is the one found by the direct inverse method algorithm, falls in the parameter space. Notably the two minima coincide with the exception of a small offset that is due to the limited resolution of the parameter space search.

The results of this analysis indicate that given a set of possible temperatures and densities for the experiment of interest, a single global minimum for the least squares fitting error can be found. The region around the global minimum has a gradient of small magnitude, meaning that the exact calculated minimum error will depend on the tolerance dictating the optimization algorithm. The fact that this region (defined arbitrarily as the region where the value of the error as defined by Equation 17 is less than the tolerance of 1×10^{-3}) spans a large range of both temperatures (2 to 19 eV) and densities (5.6×10^{16} – 11.3×10^{16} cm^{-3}) is an indicator of how difficult it is to



(a) **(b)**

Figure 6: (a) A contour plot of the least squares error versus the sought-after parameters of temperature and density. A region of minimal error is indicated by the oval region. Notably the gradient in this region is very small. (b) By calculating the gradient of the fitting error, it is evident that the global minimum found is in a shallow region, and that the actual global minimum will be dictated by the tolerance set on the fitting routine.

resolve the parameters of the Gaussian and Lorentzian when fitting to the convolution of the two profiles. Nevertheless, using the inverse method described, it is possible to resolve the FWHMs of each profile, and the corresponding physical parameters, which is a promising indicator that the technique can be further refined to decrease the uncertainty in the fitted parameters.

4 Fourier Transform Method

The properties of convolutions of functions make problems involving them convenient to analyze in the Fourier domain. Because the Voigt function is a convolution of a Gaussian and a Lorentzian profile, the problem of resolving the physical parameters of temperature and density is greatly simplified with the application of the Fourier transform, which changes improper-integral convolutions into simple products. Moreover, the Fourier transforms of both the Gaussian and Lorentzian have a simple analytical form, in which the sought parameters of the $G(\lambda)$ and $L(\lambda)$ functions appear in transformed form, $\mathscr{F}(G)$ and $\mathscr{F}(L)$ respectively.

4.1 Fourier Transforms Applied To Convolutions

Convolutions of the form described by Equation 6 can be written as the product of the Fourier transformed constituent functions.[8] Let $\mathscr{F}_G$ be the Fourier transform of the Gaussian function and $\mathscr{F}_L$ be the Fourier transform of the Lorentzian function. Taking the Fourier transform of the Voigt function V, that is the Fourier transform of the convolution of the Gaussian and Lorentzian, yields the following:

$$\mathscr{F}(V) = \mathscr{F}(G \otimes L) = \mathscr{F}_G \mathscr{F}_L. \tag{18}$$

For a Voigt profile with a centroid $\lambda_c = 0$, the Fourier transforms of the constituent functions

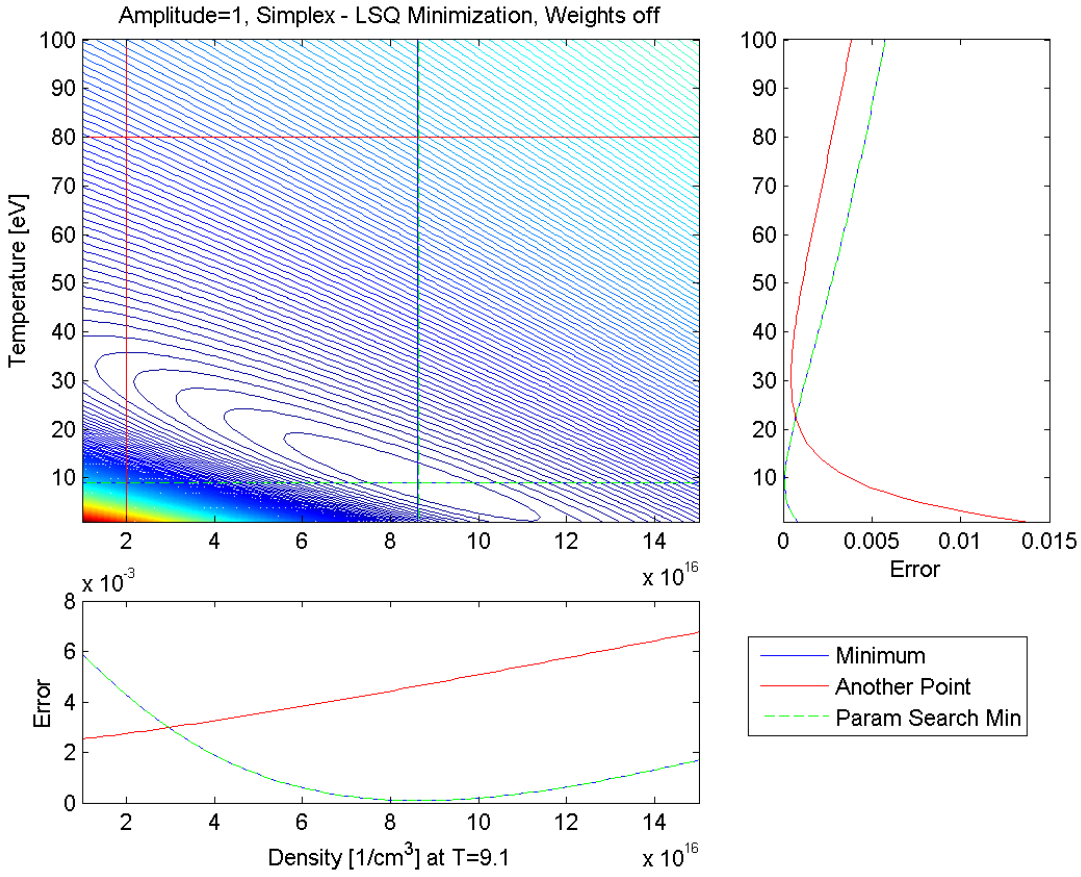


Figure 7: Contour plot of least squares error showing the location of the minimum found with the optimization routine (blue), another point (red), and the location of the minimum found in the simple parameter search (green). The respective cross sections in temperature and density domains are shown.

are purely real:

$$\mathscr{F}_G(k)\Big|_{\lambda_c=\lambda_0=0} = \frac{1}{\sqrt{8\ln 2}} G_w \exp\left(-\frac{G_w^2 k^2}{16\ln 2}\right) \tag{19}$$

$$\mathscr{F}_L(k)\Big|_{\lambda_c=\lambda_0=0} = \frac{1}{2} L_w \sqrt{\frac{\pi}{2}} \left(-\frac{1}{2} L_w |k|\right). \tag{20}$$

Spectra, however, by their very nature, are centered at finite wavelengths that are greater than zero. Shifting the centroid of a Voigt function away from zero by λ_0 corresponds to multiplication by $\exp(i\lambda_0 k)$ in the Fourier domain. As such, the analytic Fourier transforms of the Gaussian (as defined by Equation 1) and Lorentzian (as defined by Equation 3) become

$$\mathscr{F}_G(k)\Big|_{\lambda_c=\lambda_0} = \frac{1}{\sqrt{8\ln 2}} G_w \exp\left(i\lambda_0 k\right) \exp\left(-\frac{G_w^2 k^2}{16\ln 2}\right) \tag{21}$$

$$\mathscr{F}_L(k)\Big|_{\lambda_c=\lambda_0} = \frac{1}{2} L_w \sqrt{\frac{\pi}{2}} \exp\left(i\lambda_0 k\right) \exp\left(-\frac{1}{2} L_w |k|\right). \tag{22}$$

where λ_0 denotes the centroid, k denotes wave number in the Fourier domain, and G_w and L_w are defined in the same way as in Equation 2 and Equation 5 respectively. Notably this form (Equation 21 and Equation 22) is consistent with the form seen in Equation 19 and Equation 20 since $\exp(i\lambda_0 k) = 1$ if the centroid is at zero. This means that all information about the shape of the Voigt profile is preserved in the real part of $(\mathscr{F}_G \mathscr{F}_L)$ and the imaginary part is purely related to the shift of the centroid away from zero. The real part, which in this case coincides with the magnitude, can be obtained by multiplying $(\mathscr{F}_G \mathscr{F}_L)$ by its complex conjugate and taking the square root.

Thus if Voigt-like light intensity data is given by the set of values $\{I_i\}$, then the fast Fourier transform of $\{I_i\}$ – that is $\mathscr{F}\{I_i\}$ – should be well-approximated by the product of $\mathscr{F}_G$ and $\mathscr{F}_L$. Ideally, parameters G_w and L_w could be found such that $\mathscr{F}\{I_i\}$ would be exactly equal to $\mathscr{F}_G \mathscr{F}_L$ – the real and imaginary components would match. However, because data is imperfect and the Voigt function is an estimate of the profile of a true spectral line, there is an error such that in actuality

$$\mathscr{F}\{I_i\} = (\mathscr{F}_G \mathscr{F}_L) + error \tag{23}$$

where $\mathscr{F}\{I_i\}$, $\mathscr{F}_G$, $\mathscr{F}_L$, and $error$ are all complex-valued. The error becomes more pronounced when incorporating the centroid shift. This is because any asymmetry in $\{I_i\}$ contributes to an artificial shift that magnifies the imaginary part of the error.

Notably, in order to extract temperature and density (corresponding to G_w and L_w respectively), it is not necessary to fit to the centroid λ_0 and hence the imaginary part of $\mathscr{F}\{I_i\}$ is inconsequential, meaning it has nothing to do with G_w and L_w. As discussed previously, information about the shape of the Voigt profile is contained entirely in $\mathcal{R}e(\mathscr{F}_G \mathscr{F}_L)$. This is particularly advantageous in that by fitting in the Fourier domain, one parameter – the centroid – is eliminated, while preserving all other relevant parameters. By eliminating one parameter, it is possible that fitting in the Fourier domain could yield a more accurate fit, that is a more accurate value of temperature and density. To test this hypothesis, the Fourier transform fitting method was implemented.

4.2 Fitting Routine in the Fourier Domain

An algorithm was constructed to determine the parameters of interest (T and n_e) by fitting in the Fourier domain. In order to determine the accuracy of this fit, a simulated intensity dataset

with coordinates ($\{\lambda_i\}_{sim}$,$\{I_i\}_{sim}$) was constructed so that the simulated temperature and density were known a priori. A simulated intensity data set, without random noise, can be constructed as follows:

1. Select a centroid λ_0 and a set of wavelength coordinates $\{\lambda_i\}_{sim}$ centered at λ_0.
2. Select a temperature T_{sim} value and a density $n_{e,sim}$ value that is representative of the actual spectral data.
3. Input T_{sim} and $n_{e,sim}$ values into Equation 2 and Equation 5 respectively to determine L_w and G_w.
4. Input G_w and L_w into expressions for the Gaussian and Lorentzian (Equation 1 and Equation 3 respectively).
5. Convolve the Gaussian and Lorentzian functions using the discrete form of the convolution (see Equation 12).
6. The result is a simulated Voigt profile ($\{\lambda_i\}_{sim}$,$\{I_i\}_{sim}$) that represents a spectral line profile.

By taking the fast Fourier transform of the simulated data, the $\{I_i\}_{sim}$ dataset is transformed into the Fourier domain. A least-squares fit is then performed in the Fourier domain by optimizing the parameters G_w and L_w (contained in the product $\mathscr{F}_G\mathscr{F}_L$) so as to minimize the error term in Equation 23. It was found that optimizing the parameters G_w and L_w, and subsequently resolving T and n_e from Equation 2 and Equation 5 was more appropriate than finding T and n_e directly. The reason for this is that the disparity in the values of temperature and density is about 16 orders of magnitude, whereas the widths G_w and L_w are of the same order of magnitude. Optimization routines tend to function better when the parameters of interest are of the same order of magnitude.

The algorithm that generates fits in the Fourier domain was constructed as follows:

1. Transform data into the Fourier domain.
   (a) Take the fast Fourier transform of the intensity data $\{I_i\}$. This is performed using a built-in function in MATLAB.
   (b) Transform the wavelength data $\{\lambda_i\}$ into the Fourier domain by relating the wavenumber $\{k_j\}$ in Fourier space to the wavelength $\{\lambda_i\}$. In particular, $\{k_j\}$ is centered around zero, has the same number of elements as $\{\lambda_i\}$, and has a spacing that is related to the range of $\{\lambda_i\}$:

$$k_{n+1} - k_n = \frac{2\pi}{\lambda_{max} - \lambda_{min}}. \tag{24}$$

2. Express the analytic product of $\mathscr{F}_G$ and $\mathscr{F}_L$ in terms of unknown parameters z_1, z_2, and z_3, which correspond to G_w, L_w, and the amplitude respectively.

$$\mathscr{F}_G\mathscr{F}_L = z_3 \exp\left(-\frac{z_1^2 k^2}{16\ln 2}\right)\sqrt{\frac{\pi}{2}}\left(-\frac{1}{2}z_2|k|\right) \tag{25}$$

3. Perform a least-squares fit of $\mathscr{F}_G\mathscr{F}_L$ to $\mathscr{F}\{I_i\}$ by optimizing the parameters z_1 and z_2 (or equivalently G_w and L_w) so that the least squares error E between $\mathscr{F}_G\mathscr{F}_L$ and $\mathscr{F}\{I_i\}$ is minimized:

$$E = \sum_{i=1}^{n} \left(\mathscr{F}\{I_i\} - [\mathscr{F}_G\mathscr{F}_L]_i\right)^2. \tag{26}$$

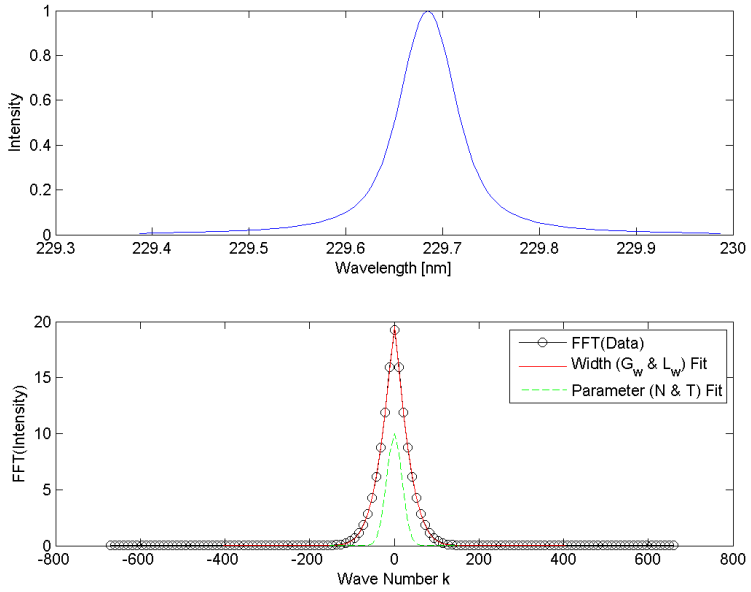


Figure 8: Top plot shows a simulated intensity profile $\{I_i\}$ that is constructed by taking temperature $T = 10$ eV, and density $N = 8 \times 10^{16}$ cm^{-3}. The bottom plot shows: the $\mathscr{F}\{I_i\}$ – the fast Fourier transform of the simulated Voigt profile data; a fit to $\mathscr{F}\{I_i\}$ using G_w and L_w as the sought-after parameters; and a fit to $\mathscr{F}\{I_i\}$ using temperature and density as the sought-after parameters. Notably due to the difference in the magnitude of temperature and density, the latter fit is not as good as the former.

4. Using procured values of G_w and L_w (corresponding to z_1 and z_2 respectively), determine temperature T and density n_e using Equation 2 and Equation 5 respectively.

The fit given by this algorithm for a simulated data set is shown in Figure 8, which shows a data set, the transformed data set, and the resulting fits in the Fourier domain. As an aside, G_w and L_w are better optimizing parameters in this case because they are of the same order of magnitude. Whether this is a problem or not is determined by whether an optimization algorithm uses normalized tolerance, that is tolerances that are normalized by the values of interest, or unnormalized tolerances – that is the same tolerance for both parameters. Since the direct inverse method used normalized tolerances, the order of magnitude problem did not appear.

A parameter search similar to the one discussed in Section Section 3.2 was implemented for the Fourier fit to determine how the quality of the fit changed if the parameters were altered. The results of the parameter search are shown in Figure 9. The fit in the Fourier domain is compared to the actual values used to construct the simulated data. Again it is evident that the optimal solution is guided largely by the tolerance since the region of the global minimum is rather large.

4.3 Comparison to Direct Inverse Method

The direct inverse method discussed in Section 3 was applied to the same simulated data as that used to test the Fourier method. This simulated data is shown in the top plot of Figure 8. The temperature and density used to generate the simulated data were 10 eV and 8×10^{16} cm^{-3} respectively. The direct inverse method was able to resolve the temperature and density much better than the Fourier transform method. The parameter search results are shown in Figure 10, which can be compared to the results of the Fourier method (Figure 9). The inverse method output

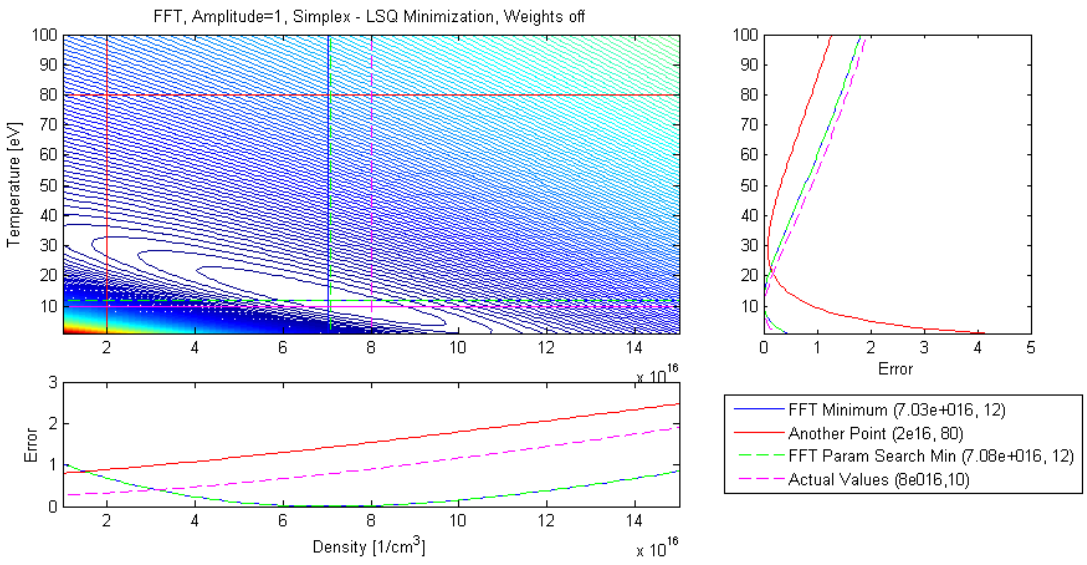


Figure 9: A parameter space search was performed to determine how the optimal values found using the Fourier transform fitting algorithm compared to other combination of temperature and density values. The blue cross indicates the optimum value found by the algorithm, the green cross indicates the optimum value found by the parameter space search (which is slightly different from the optimal value on account of parameter space discretization), and the magenta cross indicates the values that were used to construct the initial dataset $\{I_i\}$. Notably, the parameters found through the Fourier method do not coincide with the actual temperature and density values.

a temperature of 9.85 eV and a density of 7.999×10^{16} cm^{-3}, values that are very close to the actual values. Conversely, the Fourier method yielded a temperature of 12 eV and a density of 7.03×10^{16} cm^{-3}, values that are not as close to the actual.

Figure 9 and Figure 10 also illustrate consistency between the two methods of parameter optimization. That is, the two methods find global minimum regions that are very close to each other in size and scope. Because the region around the minimum is so large (in terms of ranges of parameter space variables) for both cases, it is not immediately clear, which method – the direct inverse or the Fourier method – is more robust. In order to test the resolving capabilities of the two methods, the validity of each method should be tested with a variety of datasets. The implementation of such evaluations, however, falls outside the scope of this paper, the focus of which is to show that temperature and density can be resolved by deconvolving spectral lines into their constituent profiles.

5 Conclusions

Two methods for deconvolving spectral profiles have been formulated, computationally implemented, and refined: a direct inverse method and an inverse method involving Fourier transforms. These methods both yield the sought-after parameters of temperature and density, which are associated with the constituent functions that form the convolved profile. Because these constituent functions are – by nature – similar in shape, resolving the exact values of the FWHMs, and hence the temperature and density, is difficult. This difficulty is demonstrated by doing a parameter space search to determine how a fit is affected by changing the parameter values away from the optimal values determined via least-squares fitting routines. It was found that a direct inverse method is better able to resolve temperature and density values for the case of simulated spectral profiles.

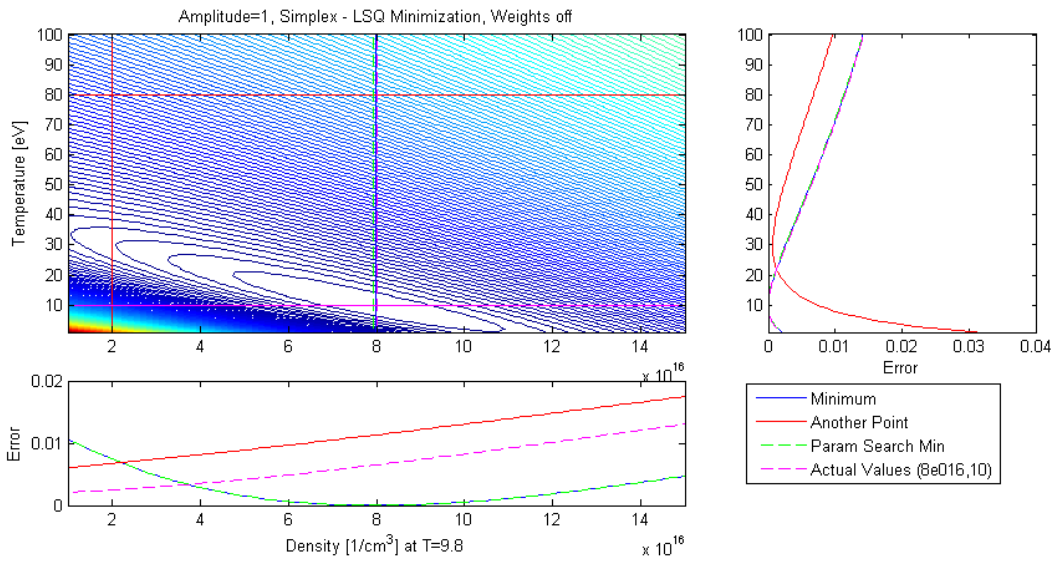


Figure 10: The direct inverse method is better able to resolve temperature and density based on a least-squares fit made on simulated data whose temperature and density, and hence constituent Gaussian and Lorentzian profiles are known. This is evidenced by the fact that the optimization minimum (blue), parameter search minimum (green), and actual values (magenta) all coincide.

Nevertheless, the two methods are consistent in their ability to find the region of minimum error in the T-n_e parameter space.

The robustness of each method can be further tested by applying the methods to noisy simulated profiles – that is Voigt profiles with random error that would more-accurately simulate data and inherent noise. Further investigation of how the accuracy of the methods changes when G_w and L_w are comparable, versus when they are not would also shed light on the applicability of each method. In effect, investigating how the algorithms behave for the case of different simulated datasets – such as different combinations of temperature and density, varied noise levels, varied tolerances for convergence, etc – would further illustrate the range of applicability of each method and allow for further refinement of the algorithms. Such analysis will be implemented in the near future, but is beyond the scope of this report, the contents of which are intended to demonstrate the feasibility of two deconvolution methods as applied to spectral data.

Acknowledgements

The author would like to thank the RTG Grant for its support and funding of this research as well as Professor James Morrow of the Mathematics Department and Professor Uri Shumlak of the Aeronautics and Astronautics Department for their invaluable mentorship and guidance on this project.

References

[1] V. Milosavljevic, G. Poparic, *Atomic spectral line free parameter deconvolution procedure*, Physical Review E 63, 2001 John A Gubnert, *A new series for approximating Voigt functions, IOP Journal of Physics A: Mathematical and General*, 27, 1994.

[2] Jian He and Chunmin Zhang, *The accurate calculation of the Fourier transform of the pure Voigt function*, Journal of Optics A: Pure and Applied Optics, 2005.

[3] A.B. McLean, C.E.J. Mitchell, D.M. Swanston, *Implementation of an efficient analytical approximation to the Voigt function for photoemission lineshape analysis*, 2002.

[4] Yuyan Liu, Jieli Lin, Guangming Huang, Yuanqing Guo, Chuanxi Duan, *Simple empirical analytical approximation to the Voigt profile*, J. Opt. Soc. Am. B 18, 666-672 (2001).

[5] A. Ionascut-Nedelcescu, C. Carlone, U. Kogelschatz, D.V. Gravelle, and M.I. Boulos, *Calculation of the gas temperature in a throughflow atmospheric pressure dielectric barrier discharge torch by spectral line shape analysis*, Journal of Applied Physics 103, 063305 March 2008.

[6] I.H. Hutchinson, Principles of Plasma Diagnostics 2nd Ed., Cambridge University Press, Cambridge, 2002.

[7] H.R. Griem, Spectral Line Broadening By Plasmas, Academic Press Inc., New York, 1974.

[8] P.A. Jansson, Deconvolution with Applications in Spectroscopy, Academic Press Inc., Orlando, 1984.

[9] R.P. Golingo, *Formation of a Sheared Flow Z-Pinch*, University of Washington PhD Dissertation, 2003.

[10] Ralchenko, Yu., Kramida, A.E., Reader, J., and NIST ASD Team (2008). NIST Atomic Spectra Database (version 3.1.5), [Online]. Available: `http://physics.nist.gov/asd3` [2010, May 15]. National Institute of Standards and Technology, Gaithersburg, MD.

[11] S. S. Harilal, C. V. Bindhu, Riju C. Issac, V.P.N. Nampoori, and C.P.G. Vallabhan. *Electron density and temperature measruements in laser prdouced carbon plasma*, Journal of Applied Physics Vol. 82 No. 5, September 1997

[12] Yuyan Liu, Jieli Lin, Guangming Huang, Yuanqing Guo, Chuanxi Duan. *Simple empirical analytical approximation to the Voigt profile*, Journal of the Optical Society of America, Vol. 18 No. 5, May 2001

[13] D. Salzman. Atomic Physics in Hot Plasmas, Oxford University Press Inc., New York, 1998. pg 168–183

[14] Lagarias, J.C., J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions," SIAM Journal of Optimization, Vol. 9 Number 1, pp. 112-147, 1998.

A Nomenclature

$\otimes$	convolution symbol, such that $g \otimes h$ is the convolution of g and h
A	amplitude of Voigt profile
c	speed of light ($3{\times}10^8$ m/s)
E	error, also written as $error$
FHWM	full width at half of the maximum of a given function
$\mathscr{F}_G$	analytic Fourier transform of a Gaussian function
$\mathscr{F}_L$	analytic Fourier transform of a Lorentzian function
$\mathscr{F}\{I_i\}$	discrete fast Fourier transform of dataset $\{I\}$
G	function describing Gaussian profile
G_{max}	amplitude of the Gaussian profile
G_w	FHWM of the Gaussian profile described by Equation 1, [nm]
I	intensity [arbitrary units]
$\{I_i\}$	set of intensity data, where (λ_i, I_i) denotes one coordinate point
$\{I_i\}_sim$	set of simulated intensity data, where (λ_i, I_i) denotes one coordinate point
I_{max}	maximum intensity
m_i	mass of radiating ion [kg]
L	function describing Lorentzian profile described by Equation 3
L_{max}	the amplitude of the Lorentzian function
L_w	the FWHM of the Lorentzian function
n	the last index of a given set
n_e	the electron number density in cm^{-3}
N	the electron number density in cm^{-3} (nomenclature used in plots)
T	temperature [eV], the product of temperature in Kelvin and Boltzmann constant 8.617×10^{-5} eV/K
T_e	the electron temperature [K]
u_i	unknown variable in inverse fitting procedure, $u_1 \rightarrow$ temperature, $u_2 \rightarrow$ density, $u_3 \rightarrow$ centroid, $u_4 \rightarrow$ amplitude
w	electron impact parameter
z_i	unknown variable in Fourier domain fit, $z_1 \rightarrow G_w$, $z_2 \rightarrow L_w$, $z_3 \rightarrow$ amplitude
α	static ion broadening parameter
λ	wavelength [nm]
λ_0	the centroid wavelength [nm]
λ_{nom}	nominal empirically determined centroid based on electron energy transition. λ_{nom} does not necessarily coincide exactly with the centroid λ_0 seen in the data
$\{\lambda_i\}$	set of wavelength data, where (λ_i, I_i) denotes one coordinate point
$\{\lambda_i\}_sim$	set of simulated wavelength data, where (λ_i, I_i) denotes one coordinate point

B Matlab Code

B.1 Optimization Function and Parameter Space Search

fitVoigt.m

```
clear all; close all;

%Input: shot number of interest.  This specifies the spectra file that is
%to be processed.  The spectra file consists of the following matrix
%|wavelength|ch1|ch2|...|ch20|.  Load C and O files seperately. This code
%calls the following functions:
%         fitVoigtfitVoigtfittingfunc - minimization procedure to find parameters of
%                         Lorentzian and Gaussian
%         fitVoigtConvolveLG  - takes temperature T, density N, mass, speed of
%                         parameters, computes Lorentzian and Gaussian
%                         widths, computes L & G and convolves the two.
%                         Amplitude is not included.
%         fitVoigtfittingfuncFFT -
shots=80313017;  % Carbon, O III
for j=1:length(shots)
    shotnumber=num2str(shots(j));
    file=strcat('ICCD_data_for_genia', shotnumber,'.txt');
    spectra(:,:,j)=load(file);
end

whichshot= ' C III 229 nm Line';             % ' O III 306 nm Line'
weights='off';
shotnum= num2str(shots);

if strcmp(whichshot, ' C III 229 nm Line')
    m=1.99442e-26;                                %mass of CIII
    center=229.687;                               %center wavelength in nm CIII (nominal)
    center=229.712;
    line=spectra(:,:,1);                          %look at one file
    wave=line(:,1);                               %pick out wavelength column
    intensity=line(:,2:21);                       %pick out intensity data
    for ch=1:20                                   %subtract background for each chord
        backgr=min(intensity(:,ch));              %background is the minimum across the 512
            pixels
        intensity(:,ch)=intensity(:,ch)-backgr; %redefine intensity
    end
    indexes=200:(length(wave)-200);               %select range of data that is of interest
    indexes=258-63:258+64;
    x=wave(indexes);                              %pick out the wavelengths
    lineIntensity=intensity(indexes,:);           %pick out the intensity
    centroidGuess=229.7;                          %centroid of the line used in fit
    axisLmin=229;                                 %axis min for graphing
    axisLmax=230.5;                               %axis max for graphing

elseif strcmp(whichshot, ' O III 306 nm Line')
    m=2.65676255e-26;                        %mass of OIII ion, kg
    center=306.3850;                         % OIV = 306.342, OIII=306.3850
    load wave0.dat; load int0.dat;
    indexes=242:280;
    x=wave0(indexes);
    lineIntensity=int0(indexes,:);
    lineIntensity=lineIntensity-ones(size(lineIntensity))*48.1;
    centroidGuess=306;
end
                                    %center wavelength in nm (arb)
                                    % W II - 252.92011
                                    % O III - 252.9327
                                    % W I - 252.9329

%Resolution Parameters
DX=0.005;                                     %denser wavelength spacing for fits
dx=mean(diff(x));
xfine=min(x):DX:max(x);                      %denser x for fits
```

```
dxfine=mean(diff(xfine));


%%Physical Parameters
W=2.8e-2;                                          %e- impact parameter angstroms,S. S. HARILAL
    , C. V. BINDHU,
c=3e8;                                             %speed of light, m/s


%% Data Voigt profile
%W=linspace(W/100, W*100, 1000);
rNf=zeros(size((W)));            %density
rTf=zeros(size((W)));            %temperature
rCf=zeros(size((W)));            %centroid
rAf=zeros(size((W)));            %amplitude

for ch=3
    j=1;                         %denotes which plasma parameter to use
    vdata=lineIntensity(:,ch);
    vdata=vdata/max(vdata);

    % Find parameters P(1), P(2), P(3), P(4) (density, temp, centroid, amplitude) to fit
        Data
    options=optimset('MaxFunEvals',1e6, 'MaxIter',1e5);
    [result fval exitflag output]=fminsearch(@(P) fitVoigtfittingfunc(m,c,W(j),x,vdata,
        center,weights,P),[14e16 100 centroidGuess 1],options); %2e16 80 100
    rNf(j)=result(1);            %density
    rTf(j)=result(2);            %temperature
    rCf(j)=result(3);            %centroid
    rAf(j)=result(4);            %amplitude
    rN=rNf(j);
    rT=rTf(j);
    rC=rCf(j);
    rA=rAf(j);
    voigtFit=rA*fitVoigtConvolveLG(W,m,c,xfine,DX,rN,rT,rC);

    % Plotting and printing
    fig1=figure;

    %adjusting paper size
    %set(gcf, 'Units', 'inches', 'PaperUnits', 'inches', ...
    %   'Position', [0, 0, 8.5,11], 'PaperPosition', [0, 0, 8.5, 11]);
    plot(x,vdata,'rx-',xfine,voigtFit,'k');
        axis([axisLmin axisLmax 0 1]);
        xlabel('Wavelength [nm]');ylabel('Intensity [arb]');
        title({strcat('Shot-',shotnum, ' (Weights-', weights,')', whichshot,...
            ': Chord ', ' ', num2str(ch)); strcat(...
            'N [cm^{-3}] = ', num2str(result(1),'%10.3e\n'),...
            ', T [eV] = ', num2str(result(2),'%10.3f\n'),...
            ', Centroid [nm] = ', num2str(result(3),'%10.3f\n'));...
            strcat('LSQError = ',num2str(fval,'%10.3e\n'))});
        legend('Data','Fit')
        sprintf('%s%e\n%s%f\n%s%f\n%s%f%s%f','Density [cm^-3]',...
        rN,'Temp [eV] ',rT, 'Centroid ',rC,'Amp ',rA)
        saveas(fig1 ,'C:\Documents and Settings\Genia Vogman\My Documents\ZaP\Stark\apr2010\
            PlottingApr30_2010\FitLine_ch3','png')
end
%}


%% Exploring the Parameters Space errorNT(n,t)
options=optimset('MaxFunEvals',1e6, 'MaxIter',1e8);
mychord=3;
lineIntensityS=lineIntensity(:,mychord)/max(lineIntensity(:,mychord));  %specific line (
    chord 3),normalized
xxdata=min(x):DX:max(x);                                %wavelength domain, high resolution
co=['r','g','m','b','k'];
```

```
p=100;                                                          %resolution
Nm=linspace(1e22,15e22,p);                                      %num density, 1/m^3
N=Nm./(10^6);                                                   %num density, 1/cm^3
T=linspace(1,100,p);                                            %Temperatue, eV
errorNT=zeros(length(N),length(T));                             %error vectors

for n=1:length(N)
    for t=1:length(T)
        V=fitVoigtConvolveLG(W,m,c,xxdata,DX,N(n),T(t),rC);
        Voigt=V;
        Voigt_interp=interp1(xxdata,Voigt,x);

        %find optimum amplitude using simplex fminsearch and compute error
        myminfun1=@(Vmax) sum(abs(Vmax*Voigt_interp-lineIntensityS).^2)/length(x);
        optV1=fminsearch(myminfun1,2,options);
        errorNT(n,t)=sum(abs(optV1*Voigt_interp-lineIntensityS).^2)/length(x);
        %lsqcurvefit function
        %myminfun1=@(Vmax,XX) Vmax*interp1(xxdata,Voigt,XX);
        %optV1=lsqcurvefit(myminfun1,1,x,lineIntensityS,.95,1.05);
   end
end


%% Contour Figures
fig2=figure
    contour(N,T,errorNT',200)
    colorbar
    xlabel('Density [1/cm^3]');ylabel('Temperature [eV]');
    title(['Amplitude=1, Simplex - LSQ Minimization, Weights ', weights]);
    hold on;
    saveas(fig2 ,'C:\Documents and Settings\Genia Vogman\My Documents\ZaP\Stark\apr2010\
        PlottingApr30_2010\invContour','png')

    %paramters from fit of Chord 3 for shot 80313017
    optimalT =rT; optimalN=rN; optC=rC;                         %optimal value found by MATLAB
        minimization function
    compareT=80; compareN=2e16;                                 %comparison values
    [minError minErrorTi]=min(min(errorNT));                    %find indices minErrorTi and
        minerrorNi of minimum error value
    [minError minErrorNi]= min(errorNT(:,minErrorTi));
    minerrorN = N(minErrorNi);                                  %density and temperature values
        at these indices
    minerrorT = T(minErrorTi);

    %epsilons
    epsilonT=.5*mean(diff(T));
    epsilonN=.5*mean(diff(N));

    %Determine indexes of parameters of interest (optimal, comparison, and
    %minimum error)
    indxTopt=find(T > optimalT-epsilonT & T <= optimalT+epsilonT);
    indxTcom=find(T > compareT-epsilonT & T <= compareT+epsilonT);
    indxTmin=find(T > minerrorT-epsilonT & T <= minerrorT+epsilonT);

    indxNopt=find( N > optimalN-epsilonN & N <= optimalN+epsilonN);
    indxNcom=find( N > compareN-epsilonN & N <= compareN+epsilonN);
    indxNmin=find( N > minerrorN-epsilonN & N <= minerrorN+epsilonN);

    %plotting
        fig3=figure
        ax1=subplot(3,3,[1 5]);
        contour(N,T,errorNT',200);
        hold on;
        plot([N(1) N(end)],[optimalT optimalT],'b');   plot([optimalN optimalN],[T(1) T(end)
            ],'b');
        plot([N(1) N(end)],[compareT compareT],'r');   plot([compareN compareN],[T(1) T(end)
            ],'r');
```

```
187         plot([N(1) N(end)],[minerrorT minerrorT],'g--'); plot([minerrorN minerrorN],[T(1) T(
                end)],'g--');
188         hold off;
189         ylabel('Temperature [eV]');
190         title(['Amplitude=1, Simplex - LSQ Minimization, Weights ', weights])
191
192         %Error vs density for T_given
193         ax2=subplot(3,3, [7 8]);
194         plot(N, errorNT(:,indxTopt));
195         hold on;
196         plot(N, errorNT(:,indxTcom),'r');
197         plot(N, errorNT(:,indxTmin),'--g');
198         hold off;
199         xlabel(strcat('Density [1/cm^3] at T=', num2str(optimalT,'%10.1f')));
200         ylabel('Error');
201
202         %Error vs temperature for
203         ax3=subplot(3,3,[3 6]);
204         plot(errorNT(indxNopt,:),T);
205         hold on;
206         plot(errorNT(indxNcom,:),T,'r');
207         plot(errorNT(indxNmin,:),T,'--g');
208         hold off;
209         xlabel('Error')
210         linkaxes([ax1 ax3],'y');
211         linkaxes([ax1 ax2],'x');
212         figure(3); legend('Minimum','Another Point','Param Search Min','Location',[.7 .2 .15
                .07]) %left bottom width height
213         saveas(fig3 ,'C:\Documents and Settings\Genia Vogman\My Documents\ZaP\Stark\apr2010\
                PlottingApr30_2010\invContourCrossec','png')
214 %}
```

B.2 Convolution Function

fitVoigtConvolveLG.m

```
1  function voigtFit = fitVoigtConvolveLG(W,m,c,rx,DX,rN,rT,rC)
2
3  Lw=(2e-16)*W*rN/10;                                          %Lorentzian width, in nm
                                                  %x2-x1
4  Gw=sqrt(8*rT*log(2)/(m*c^2))*rC*(10^(-9));                   %Gaussian width in nm
5  G=exp((-4*log(2)*(rx-rC*ones(size(rx))).^2)/(Gw^2));          %Gaussian
6  L=Lw^2./((4*(rx-rC*ones(size(rx))).^2 + Lw^2));              %Lorentzian
7  voigtFit=convn(G,L,'same')*DX;
```

B.3 Fitting Function

fitVoigtfittingfunc.m

```
1  function f = fitVoigtfittingfunc(m,c,W,x,vdata,centerNom,weights,P)
2
3  boxX=5;
4  boxY=0.1e16;
5
6  %wavelength range
7  %xx=-0.3:0.001:0.3;
8  xx=min(x):0.005:max(x);
9  %xx=xx'+centerNom.*ones(length(xx),1);
10 %lambdaC=center*ones(length(xx),1);
11
12 %evaluate lorentzian and gaussian as function
13 %of P(2)=Temperature and P(1)=density
14 %P(3)=centroid, P(4)=amplitude
15 fLw=(2e-16)^2*W^2*P(1)^2/10^2;
16 fGw=(sqrt(8*P(2)*log(2)/(m*c^2))*centerNom*(10^(-9))).^2;
17 fL= fLw./((4*(xx-P(3)).^2 + fLw));
18 fG= exp((-4*log(2)*(xx-P(3)).^2)/fGw);
```

```
19
20
21  %convolve L and G to get Voigt, that has same size as L (and G)
22  fdx=mean(diff(xx));
23  voigt=P(4)*convn(fG,fL,'same')*fdx;
24  Voigt=interp1(xx,voigt,x);
25
26
27  if strcmp(weights, 'off')
28      %Least squares fit, no weighing
29      f=sum(abs(Voigt-vdata).^2)/length(x);
30      if P(2) < 1
31          f=1e7;
32      end
33  %      if P(1) > 3.0e16 || P(1) < 3.3e16 || P(2) >11 || P(2) < 10
34  %          f=1e7;
35  %      end
36  else    %with weighing
37       for j=1:length(x)
38           if j >49 & j<70% || j > length(x)-50)  %10 - 31, or 49-70
39              a(j)=abs(Voigt(j)-vdata(j))^2;
40           else
41              a(j)=10/(Voigt(j)+10)*abs(Voigt(j)-vdata(j))^2;
42           end
43       end
44       f=sum(a)/length(x);
45       if P(2) < 1
46       f=Inf;
47       end
48  end
```